

Classificando vários clientes

Transcrição

Criamos um programa que serve para classificar uma instância.

Porém, ele não é interessante para nós, pois queremos passar um arquivo com várias instâncias. Então, basicamente o que será necessário fazer será criar um `for`, ou seja, um *loop* que percorrerá várias instâncias do arquivo.

Então, ao fim do corpo da função, escreveremos `for(int i = 0; i < dado.numInstances(); i++)`. Assim, o `i` percorrerá o número de instâncias e a cada instância percorrida, aumentará um número nesse contador. Agora, precisaremos guardar o valor da `arvore.classifyInstance(dado.instance(1))` dentro de uma variável `double` que chamaremos de `resultado`. O valor "1" era o número da instância, mas agora ele passará a ser o `i`, que aumentará conforme as repetições do laço.

Precisaremos que esse `resultado` seja um "Sim" ou "Não", pois no arquivo é dito se o número da instância que ele classificou é "0" ou "1", e não é o tipo de resposta que esperamos como saída do programa para que fique inteligível para as pessoas. Faremos um `if(resultado==0.0)` e no corpo dele haverá um `System.out("resultado da classificação:" + "yes")`. Sendo assim, quando o valor do `resultado` for "0", a saída será "yes". Criaremos uma alternativa para o "no" também, acrescentando um `else`, e então, caso o valor não seja "0", o `resultado` será "no".

Fecharemos o arquivo aberto em determinado momento durante a programação com `meuObj.close()`.

```
public class previsor {  
  
    public static void main(String[] args) throws Exception{  
  
        System.out.println("Digite o nome do arquivo:");  
  
        Scanner meuObj = new Scanner(System.in);  
  
        String nomeArquivo = meuObj.nextLine();  
  
        ArffLoader carrega = new ArffLoader();  
  
        carrega.getSource(new File(nomeArquivo));  
  
        Instances dado = carrega.getDataSet();  
  
        dado.setClassIndex(dado.numAttributes()-1);  
  
        J48 arvore = (J48) weka. core.SerializationHelper.read("j48.model");  
  
        for(int i = 0; i < dado.numInstances(); i++) {  
            double resultado = arvore.classifyInstance(dado.instance(i));  
  
            if (resultado == 0.0) {  
                System.out.println("resultado da classificação:" + "yes");  
            }  
            else {  
                System.out.println("resultado da classificação:" + "no");  
            }  
        }  
    }  
}
```

```
}  
  
}  
  
meuObj.close  
  
}
```

O nome do arquivo que rodaremos agora será `clientes.arff`. Vamos rodá-lo no terminal e será mostrado o resultado da classificação para nós. O resultado foi "yes, yes, yes" e "no, no, no". Poderemos, inclusive, abrir o arquivo e comparar se os estão acontecendo acertos. Para os 3 primeiros "yes" existem acertos, mas depois alguns erros.

De qualquer forma, já temos a saída para a classificação, um programa que funciona num terminal e serve para as pessoas que não tem o Weka instalado. Mas é possível tornar esse código ainda mais elaborado, criando soluções web ou aplicativos escritos em Java. Outros cursos da Alura poderão ser efetuados caso ainda não haja conhecimentos para isso, e assim serão usadas conjuntamente as técnicas de classificação utilizadas em Machine Learning que aprendemos.